IN4080 – 2020 FALL
NATURAL LANGUAGE PROCESSING

Jan Tore Lønning

Neural LMs, Recurrent networks, Sequence labeling, Information Extraction, Named-Entity Recognition, Evaluation

Lecture 13, 9 Nov.

Today

- Feedforward neural networks
  - Neural Language Models
- Recurrent networks
- Information Extraction
- Named Entity Recognition
- Evaluation

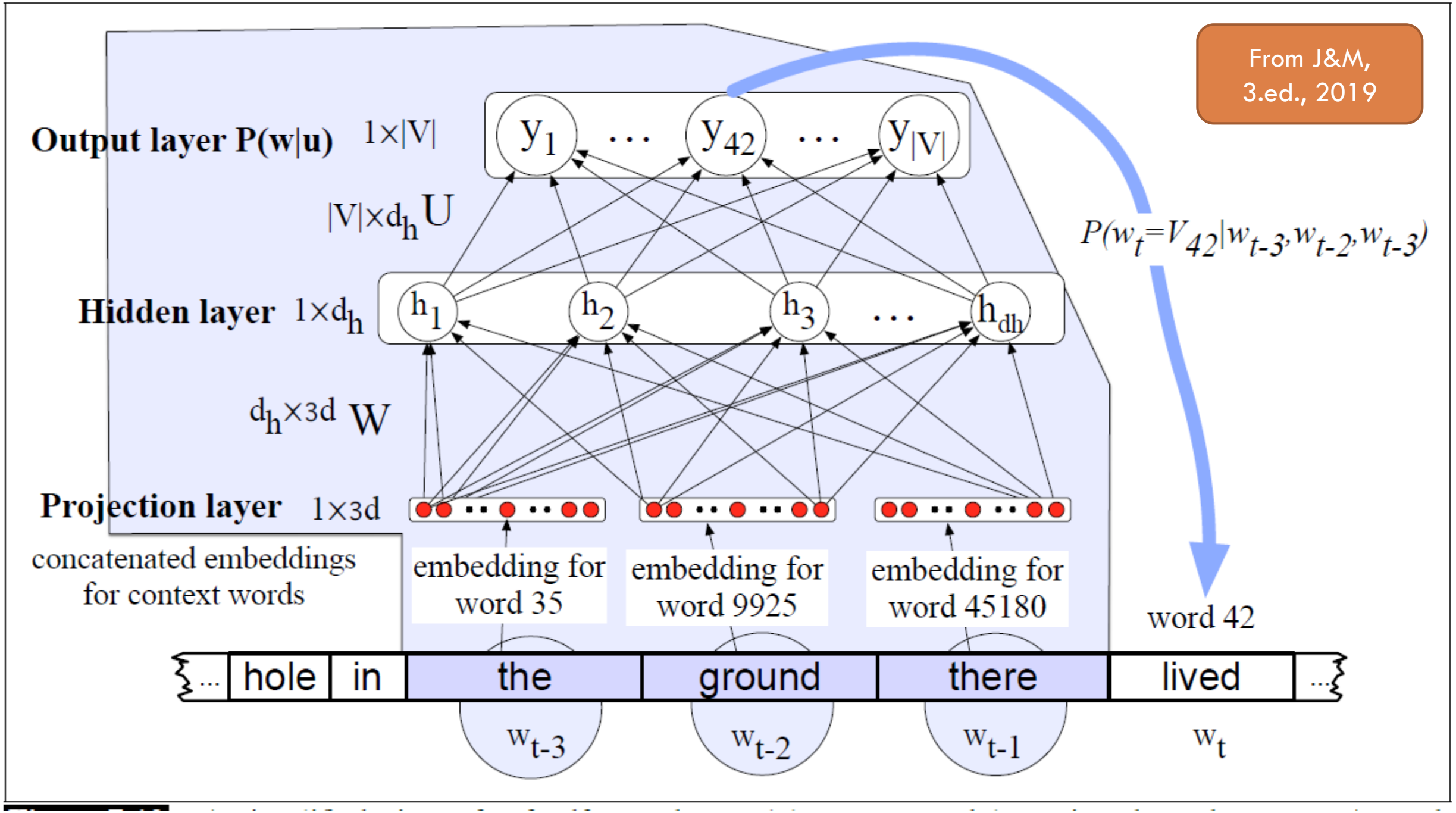
Last week

- Feedforward neural networks (partly recap)
  - Model
  - Training
  - Computational graphs
  - Neural Language Models
- Recurrent networks
- Information Extraction

Neural NLP

- (Multi-layered) neural networks
- Using embeddings as word representations

- Example: Neural language model (*k*-gram)
  - $P\left(w_i \middle| w_{i-k}^{i-1}\right)$
- Use embeddings for representing the w_i-s
- Use neural network for estimating $P\left(w_i \middle| w_{i-k}^{i-1}\right)$


From J&M, 3.ed., 2019
Output layer P(w|u)
1×|V|
y1 ... y42 ... y|V|
|V|×dh U
Hidden layer 1×dh
h1 h2 h3 ... hdh
dh×3d W
Projection layer 1×3d
concatenated embeddings for context words
embedding for word 35
embedding for word 9925
embedding for word 45180
P(wt=V42|wt-3,wt-2,wt-3)
word 42
... hole in the ground there lived ...
wt-3
wt-2
wt-1
wt

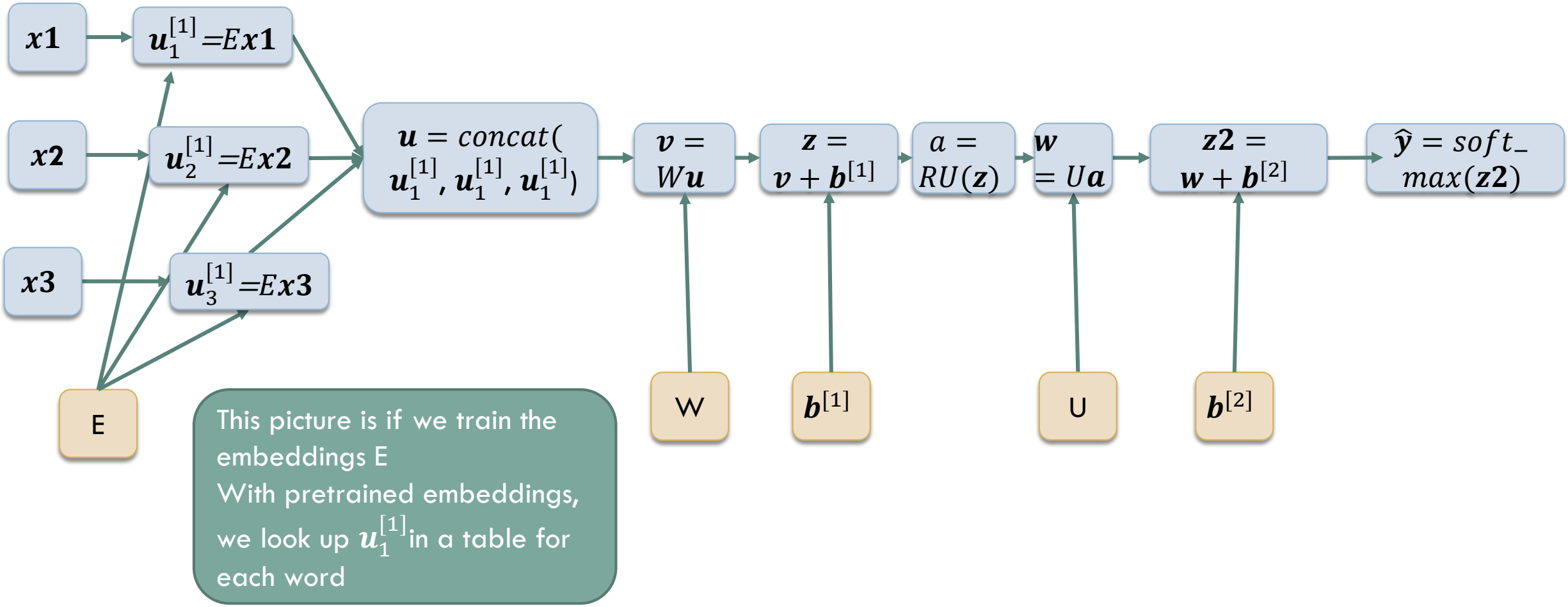
Pretrained embeddings

- The last slide uses pretrained embeddings
  - Trained with some method, SkipGram, CBOW, Glove, …
  - On some specific corpus
  - Can be downloaded from the web
- Pretrained embeddings can aslo be the input to other tasks, e.g. text classification
- The task of neural language modeling was also the basis for training the embeddings

Training the embeddings

- Alternatively we may start with one-hot representations of words and train the embeddings as the first layer in our models (=the way we trained the embeddings)
- If the goal is a task different from language modeling, this may result in embeddings better suited for the specific tasks.
- We may even use two set of embeddings for each word – one pretrained and one which is trained during the task.

Computational graph



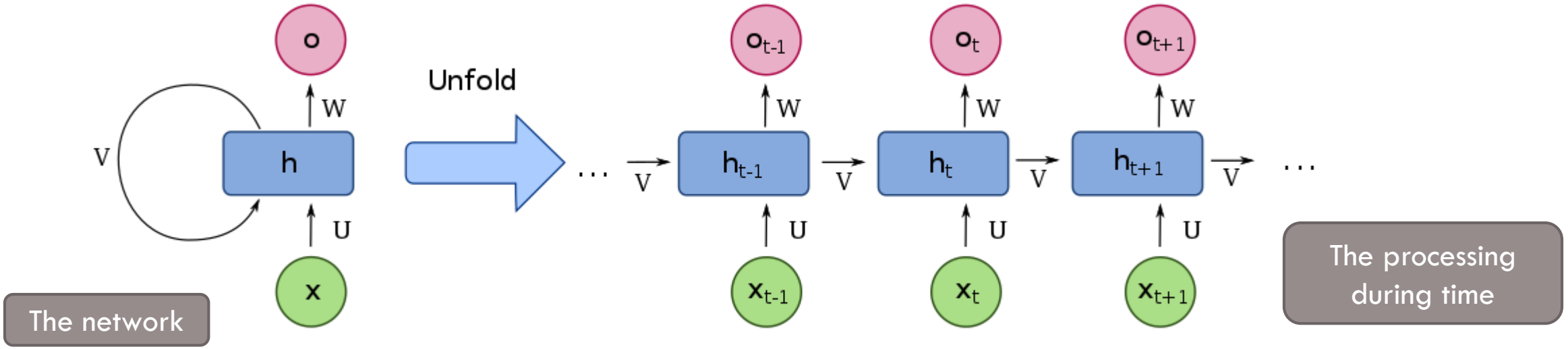
Recurrent networks

Today

- Feedforward neural networks
- Recurrent networks
  - Model
  - Language Model
  - Sequence Labeling
  - Advanced architecture
- Information Extraction
- Named Entity Recognition
- Evaluation

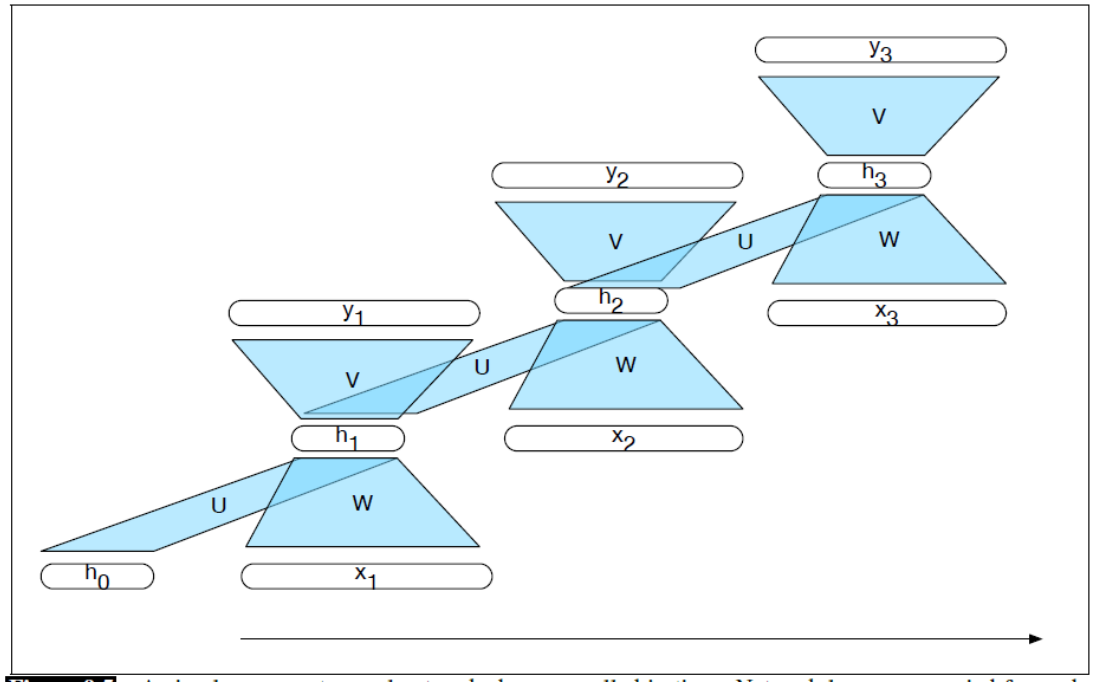
Recurrent neural nets

- Model sequences/temporal phenomena
- A cell may send a signal back to itself – at the next moment in time



https://en.wikipedia.org/wiki/Recurrent_neural_network

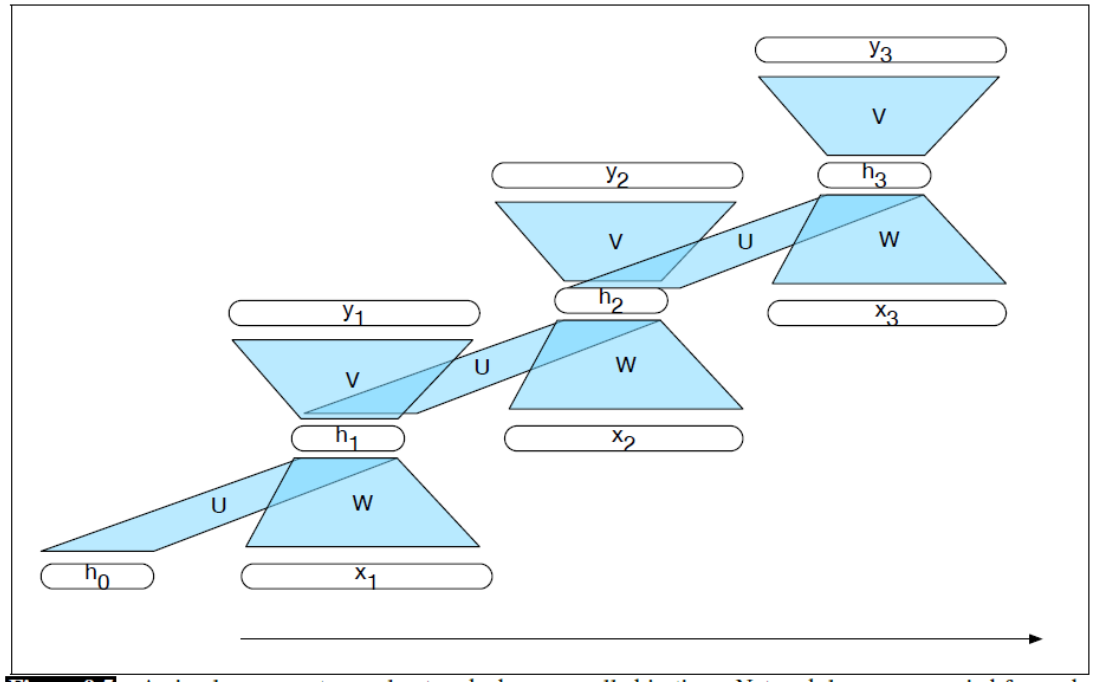
Forward



From J&M, 3.ed., 2019

- Each U, V and W are edges with weights (matrices)
- $x_1, x_2, \ldots, x_n$ is the input sequence
- Forward:
  1. Calculate h_1 from h_0 and x_1.
  2. Calculate y_1 from h_1.
  3. Calculate h_i from h_{i-1} and x_i, and y_i from i, for $i = 1, \ldots, n$

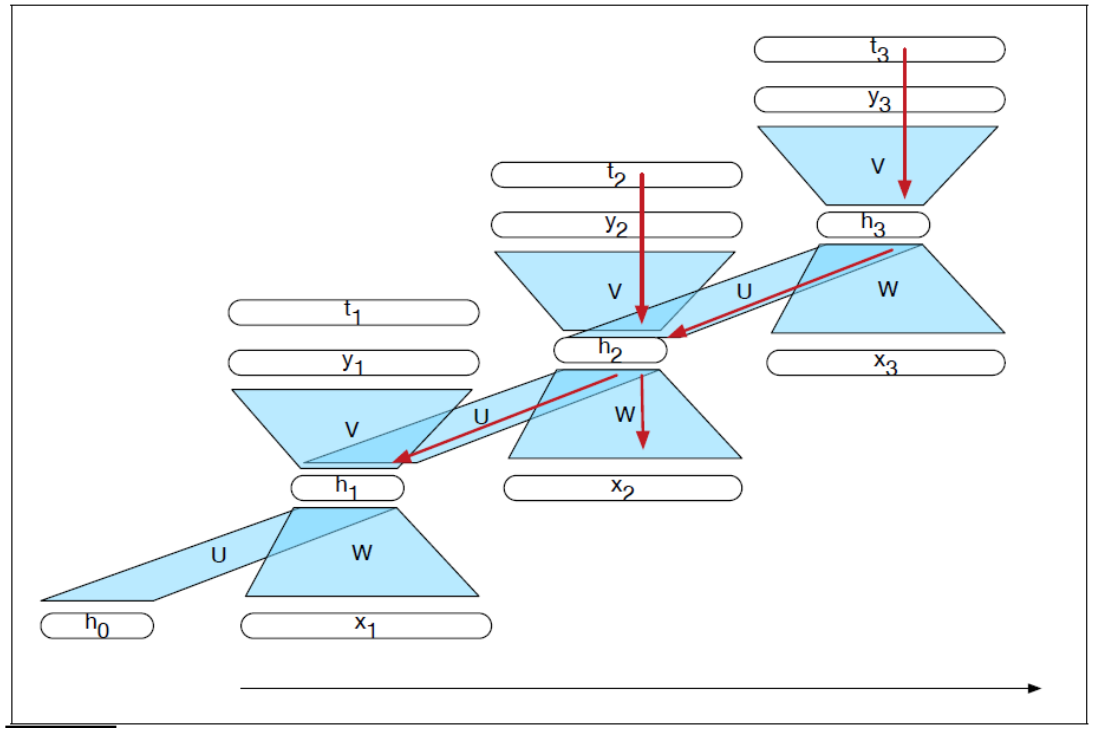
Forward



From J&M, 3.ed., 2019

- $\boldsymbol{h}_t = g(U\boldsymbol{h}_{t-1} + W\boldsymbol{x}_t)$
- $\boldsymbol{y}_t = f(V\boldsymbol{h}_t)$
- g and are activation functions
- (There are also bias which we didn't include in the formulas)

Training



From J&M, 3.ed., 2019

- At each output node:
  - Calculate the loss and the
  - δ-term
- Backpropagate the error, e.g.
  - the δ-term at $h_2$is calculated
    - from the δ-term at h_3 by U and
    - the δ-term at y_2 by V
- Update
  - V from the δ-terms at the y_i-s and
  - U and W from the δ-terms at the h_i-s

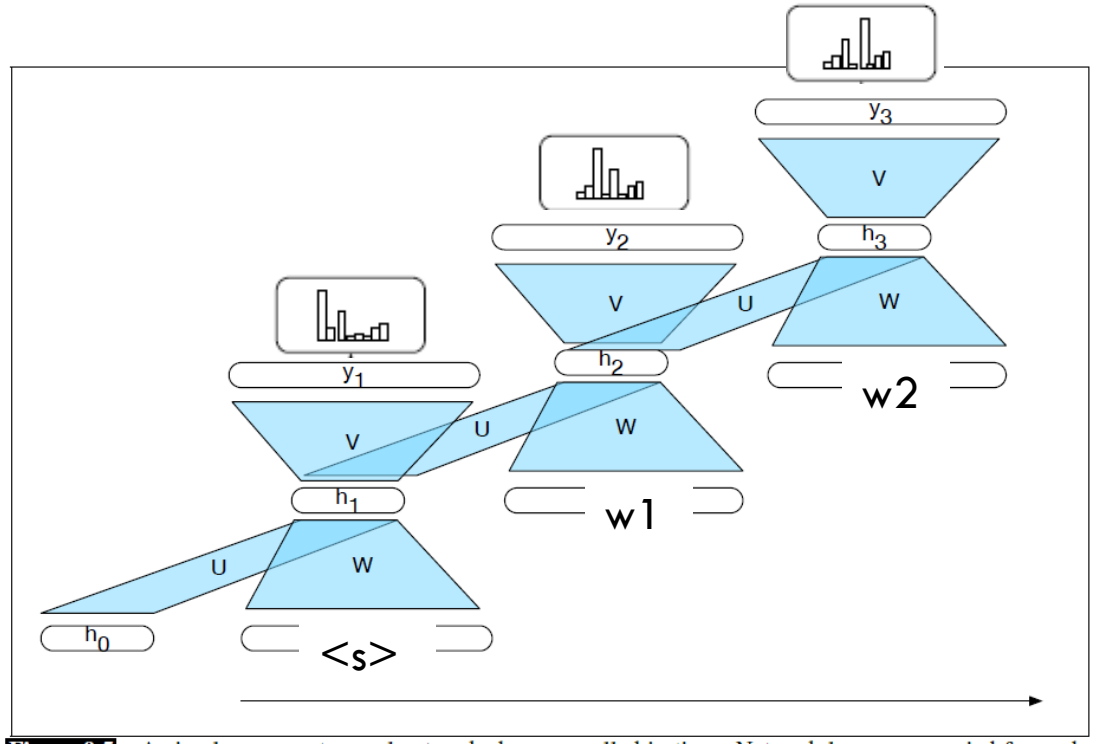
Remark

- J&M, 3. ed., 2019, sec 9.1.2 explain this at a high-level using vectors and matrices, OK
- The formulas, however, are not correct:
  - Describing derivatives of matrices and vectors demand a little more care, e.g. one has to transpose matrices
- It is beyond this course to explain how this can be done in detail
- But you should be able to do the actual calculations if you stick to the entries of the vectors and matrices, as we did above (ch. 7).

Today

- Feedforward neural networks
- Recurrent networks
  - Model
  - Language Model
  - Sequence Labeling
  - Advanced architecture
- Information Extraction
- Named Entity Recognition
- Evaluation

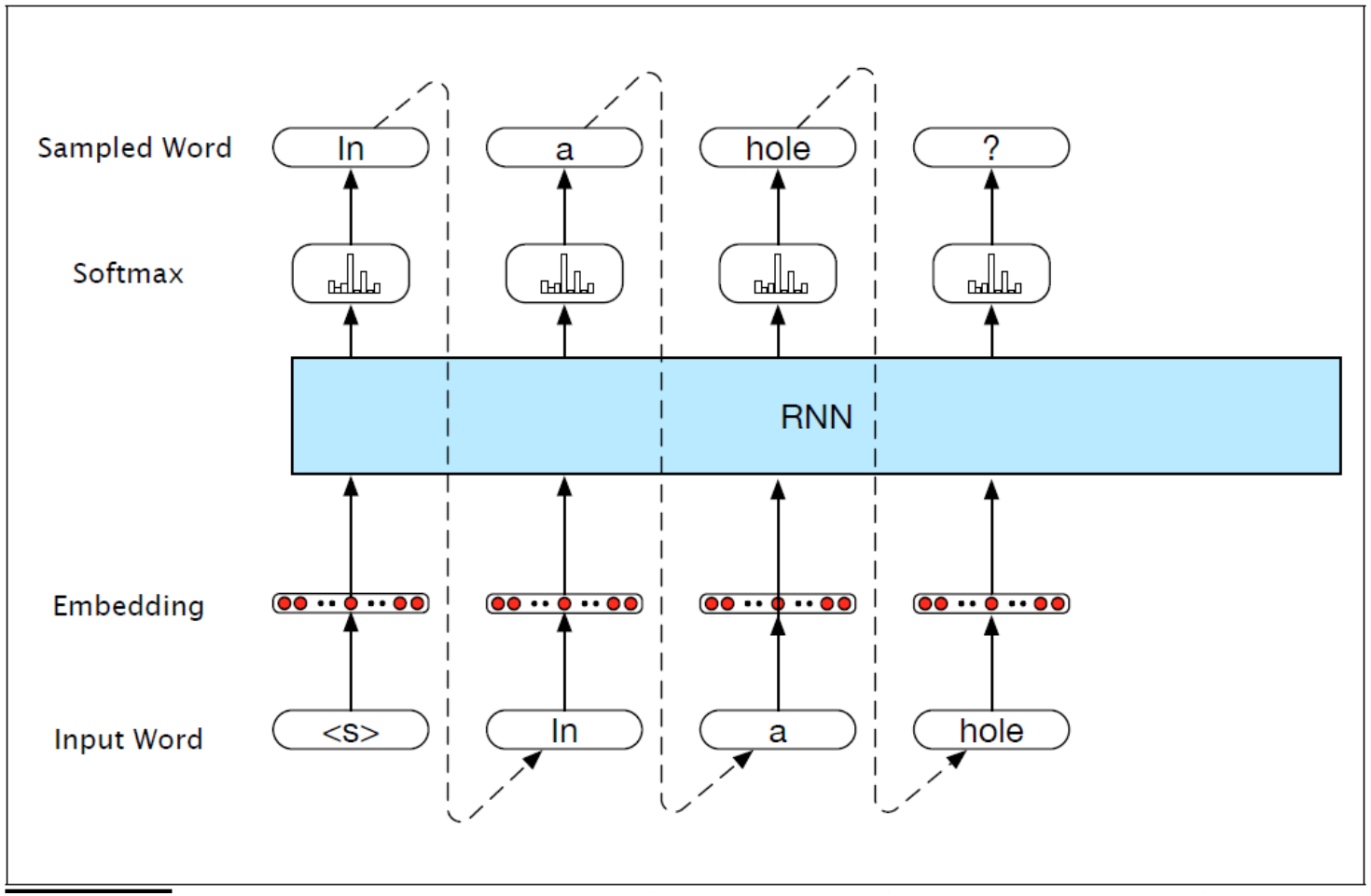
RNN Language model



From J&M, 3.ed., 2019

- $\hat{y} = P(w_n | w_1^{n-1}) = softmax(V\boldsymbol{h}_n)$
- In principle:
  - unlimited history
  - a word depends on all preceding words
- The word w_i is represented by an embedding
  - or a one-hot and the embedding is made by the LM

Autoregressive generation



From J&M, 3.ed., 2019

- Generated by probabilities:
  - Choose word in accordance with prob.distribution
- Part of more complex models
  - Encoder-decoder models
    - Translation

Today

- Feedforward neural networks
- Recurrent networks
  - Model
  - Language Model
  - Sequence Labeling
  - Sequence Labeling
  - Advanced architecture
- Information Extraction
- Named Entity Recognition
- Evaluation

Neural sequence labeling: tagging

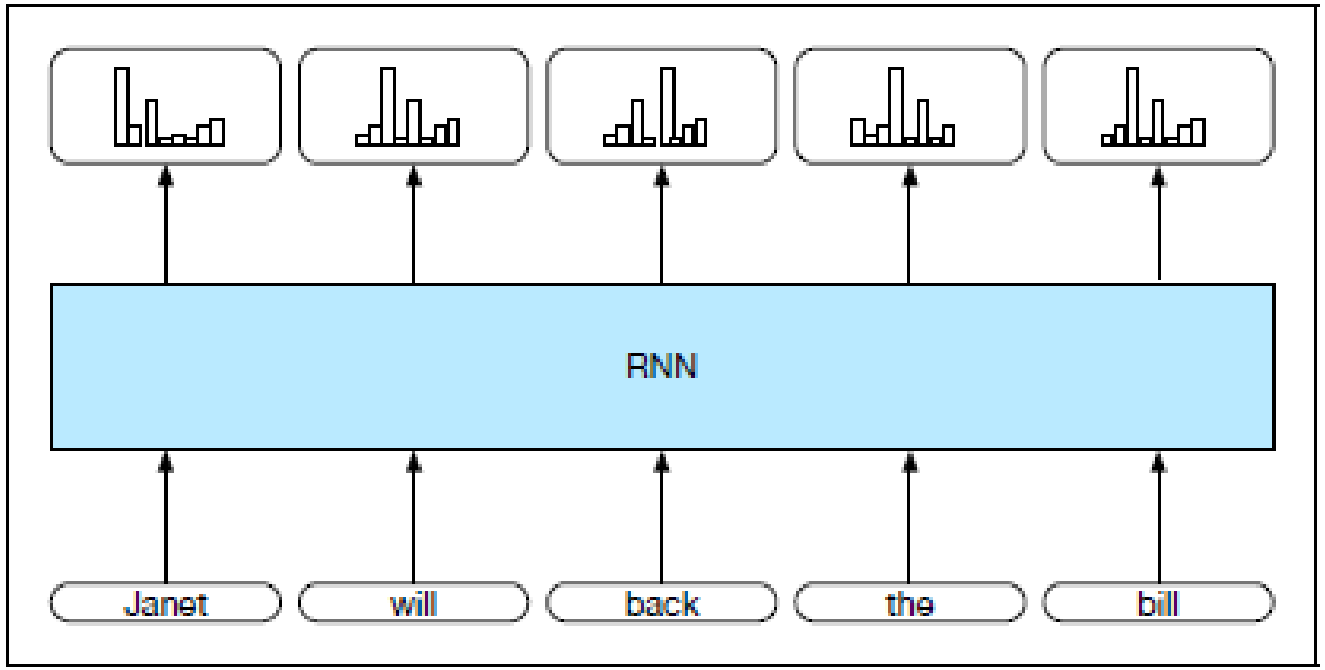


Figure 9.8 Part-of-speech tagging as sequence labeling with a simple RNN. Pre-trained word embeddings serve as inputs and a softmax layer provides a probability distribution over the part-of-speech tags as output at each time step.

From J&M, 3.ed., 2019

- $\hat{y} = P(t_n|w_1^n) = softmax(V\boldsymbol{h}_n)$

Sequence labeling

- Actual models for sequence labeling, e.g. tagging, are more complex
- For example, that it may take words after the tag into consideration.

Today

- Feedforward neural networks
- Recurrent networks
  - Model
  - Language Model
  - Sequence Labeling
  - Advanced architecture
- Information Extraction
- Named Entity Recognition
- Evaluation

Stacked RNN

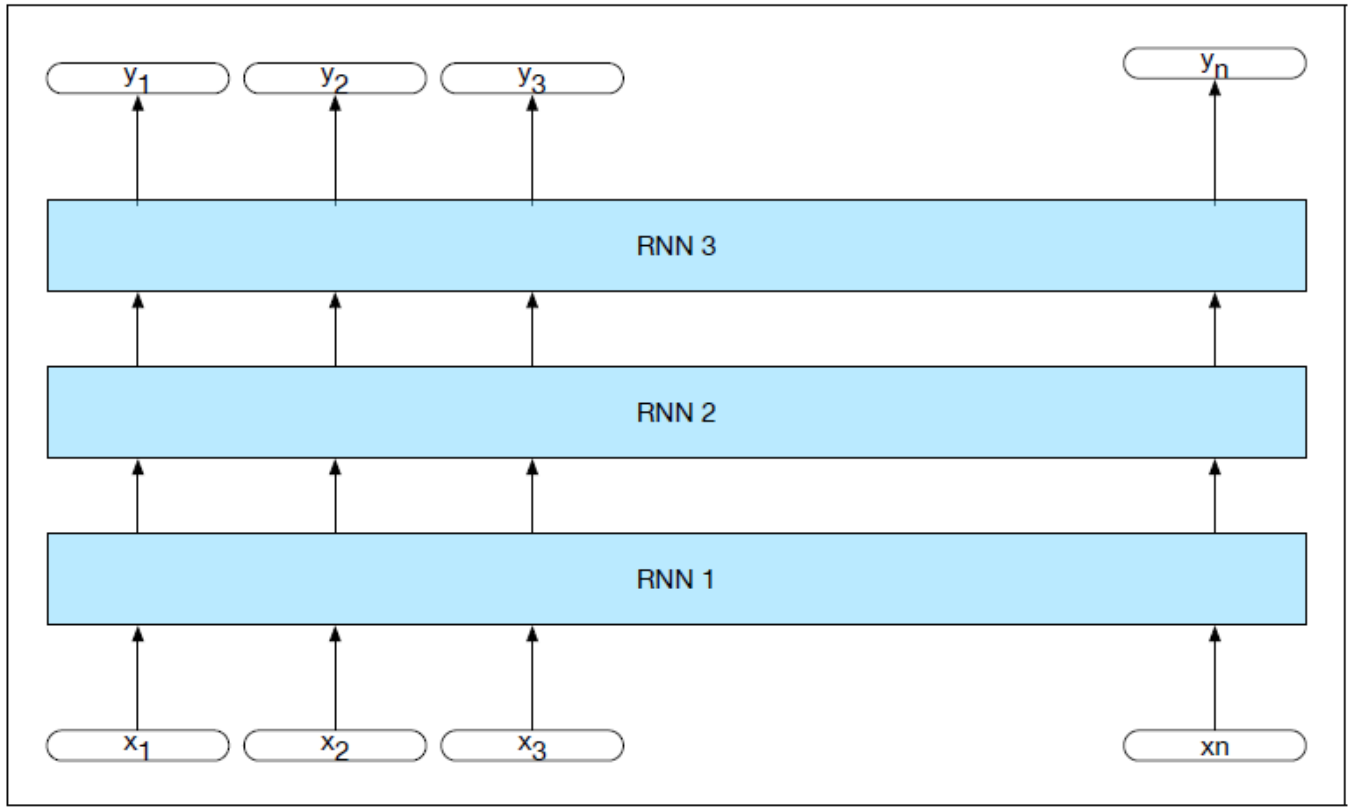


Figure 9.10 Stacked recurrent networks. The output of a lower level serves as the input to higher levels with the output of the last network serving as the final output.

From J&M, 3.ed., 2019

- Can yield better results than single-layers
- Reason?
  - Higher-layers of abstraction
    - similar to image processing (convolutional nets)

Bidirectional RNN

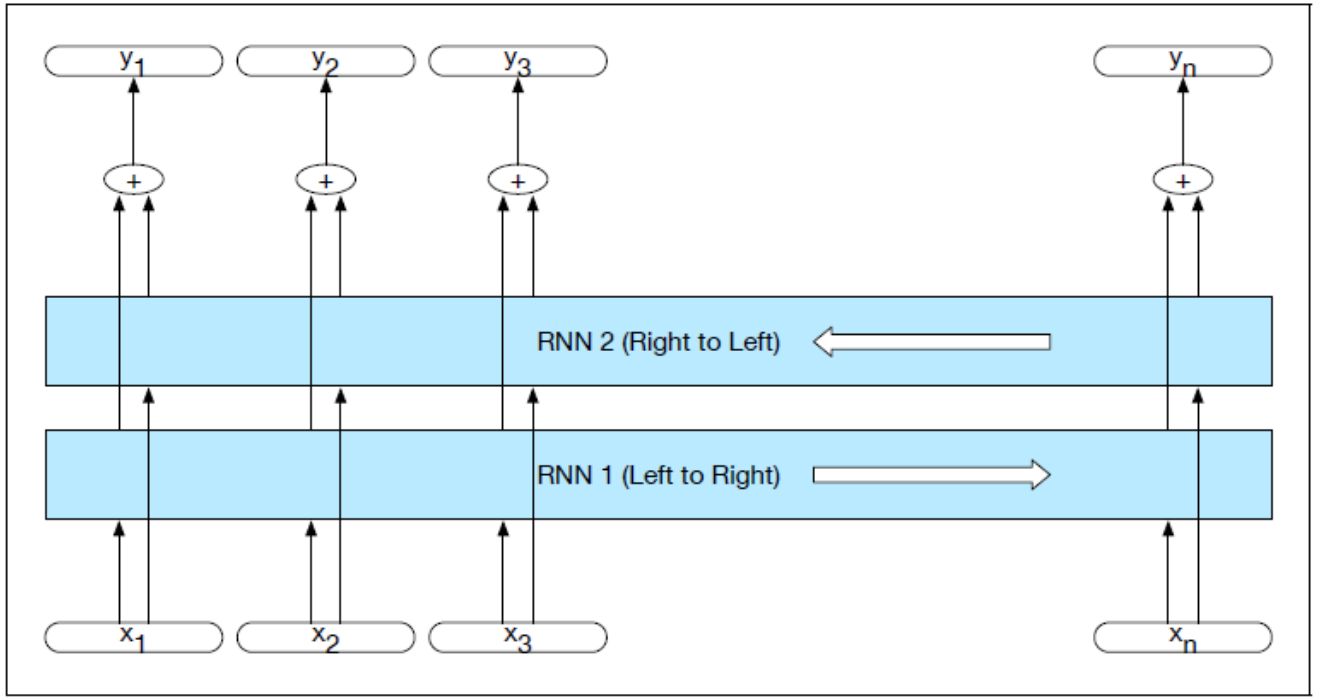


Figure 9.11 A bidirectional RNN. Separate models are trained in the forward and backward directions with the output of each model at each time point concatenated to represent the state of affairs at that point in time. The box wrapped around the forward and backward network emphasizes the modular nature of this architecture.

From J&M, 3.ed., 2019

- Example: Tagger
- Considers both preceding and following words

LSTM

- Problems for RNN
  - Keep track of distant information
  - Vanishing gradient
    - During backpropagation going backwards through several layers, the gradient approaches 0
- Long Short-Term Memory
  - An advanced architecture with additional layers and weights
    - Not consider the details here
- Bi-LSTM (Binary LSTM)
  - Popular standard architecture in NLP

Information extraction

Today

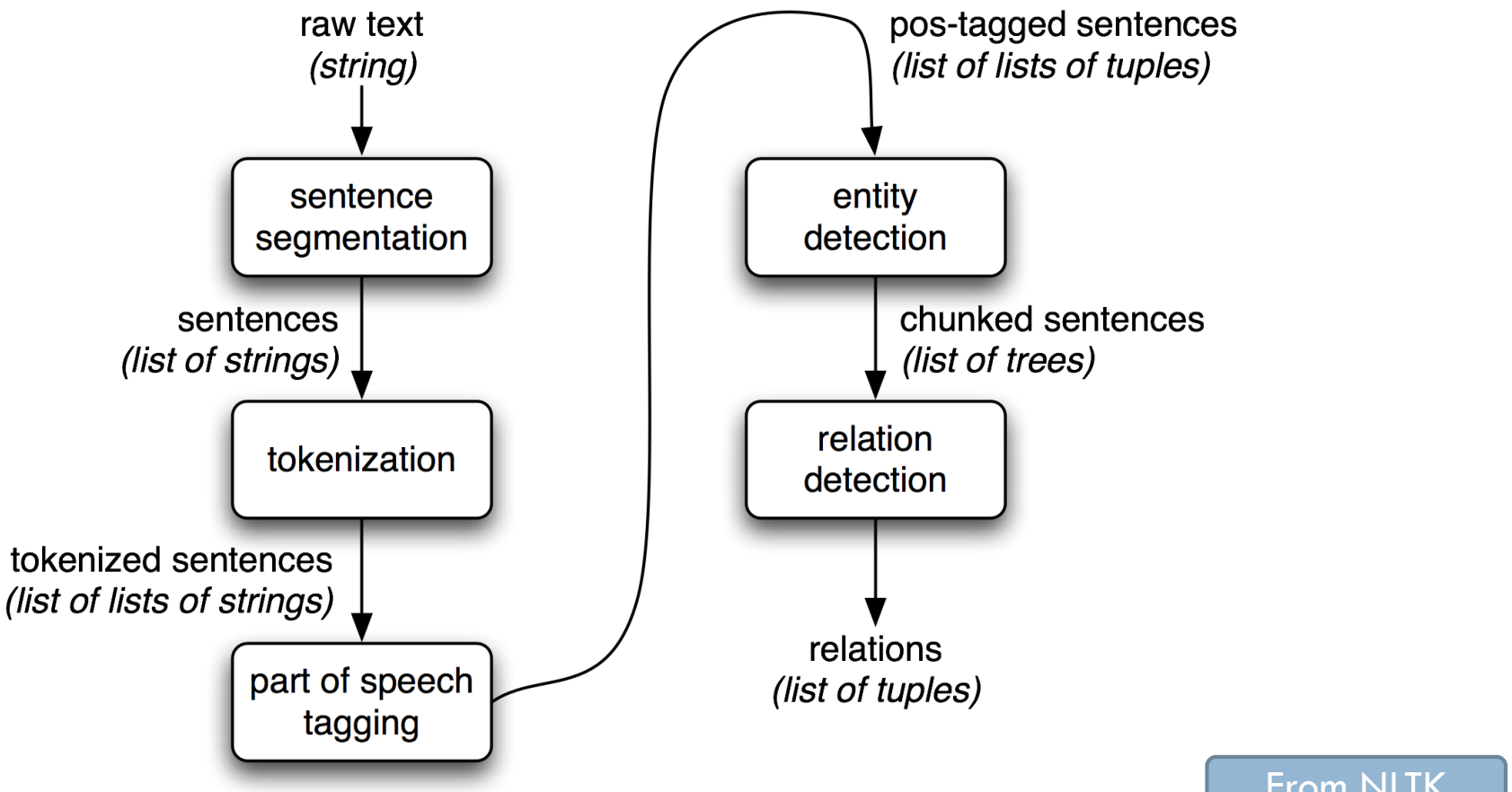
- Feedforward neural networks (partly recap)
- Recurrent networks
- Information extraction, IE
  - Chunking
- Named Entity Recognition
- Evaluation

IE basics

Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. (Wikipedia)

- Bottom-Up approach
- Start with unrestricted texts, and do the best you can
- The approach was in particular developed by the Message Understanding Conferences (MUC) in the 1990s
- Select a particular domain and task

A typical pipeline



From NLTK

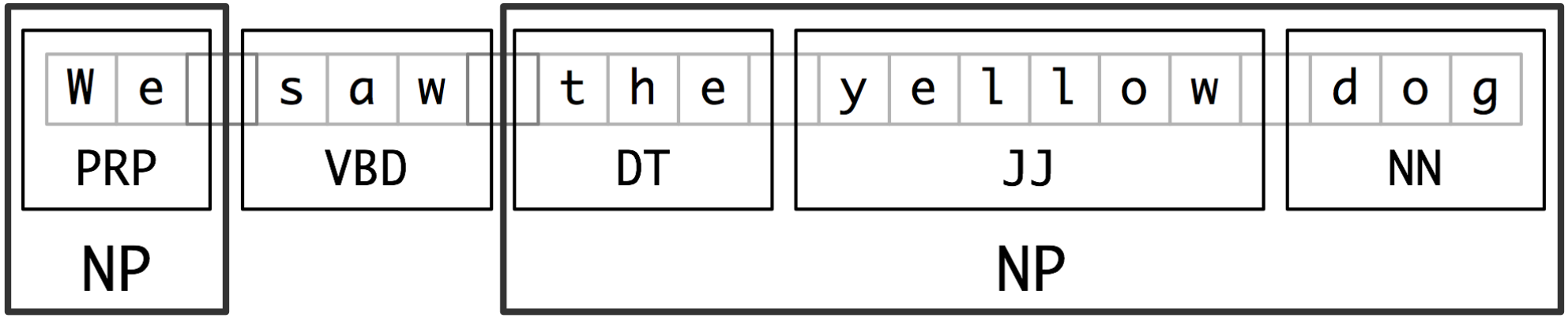
Some example systems

- Stanford core nlp: http://corenlp.run/
- SpaCy (Python): https://spacy.io/docs/api/
- OpenNLP (Java): https://opennlp.apache.org/docs/
- GATE (Java): https://gate.ac.uk/
  - https://cloud.gate.ac.uk/shopfront
- UDPipe: http://ufal.mff.cuni.cz/udpipe
  - Online demo: http://lindat.mff.cuni.cz/services/udpipe/
- Collection of tools for NER:
  - https://www.clarin.eu/resource-families/tools-named-entity-recognition

Today

- Feedforward neural networks (partly recap)
- Recurrent networks
- Information extraction, IE
  - Chunking
- Named Entity Recognition
- Evaluation

Next steps



- Chunk together words to phrases

NP-chunks

[The/DT market/NN] for/IN
[system-management/NN software/NN] for/IN
[Digital/NNP]
['s/POS hardware/NN] is/VBZ fragmented/JJ enough/RB that/IN
[a/DT giant/NN] such/JJ as/IN
[Computer/NNP Associates/NNPS] should/MD do/VB well/RB there/RB ./.

- Exactly what is an NP-chunk?
- It is an NP
- But not all NPs are chunks

- Flat structure: no NP-chunk is part of another NP chunk
- Maximally large
- Opposing restrictions

Chunking methods

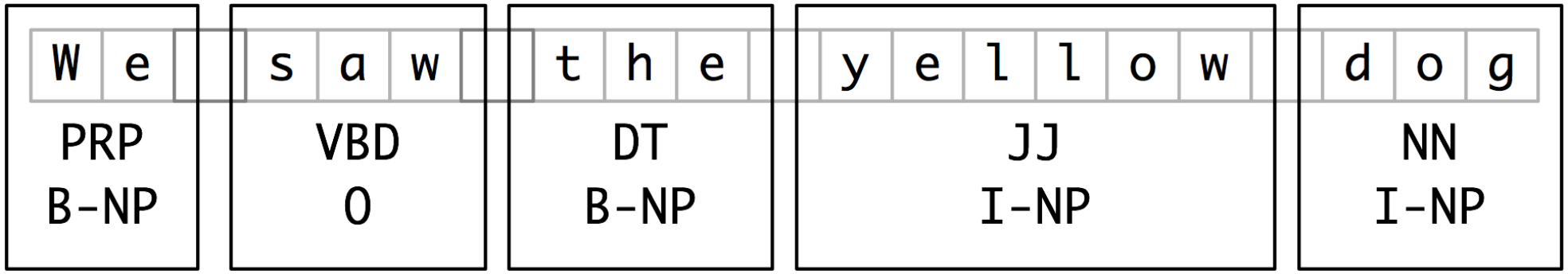
- Hand-written rules
- Regular expressions
- Supervised machine learning

Regular Expression Chunker

- Input POS-tagged sentences
- Use a regular expression over POS to identify NP-chunks
- NLTK example:
- It inserts parentheses

```
grammar = r"""
  NP: {<DT|PP\$>?<JJ>*<NN>}
      {<NNP>+}
"""
```

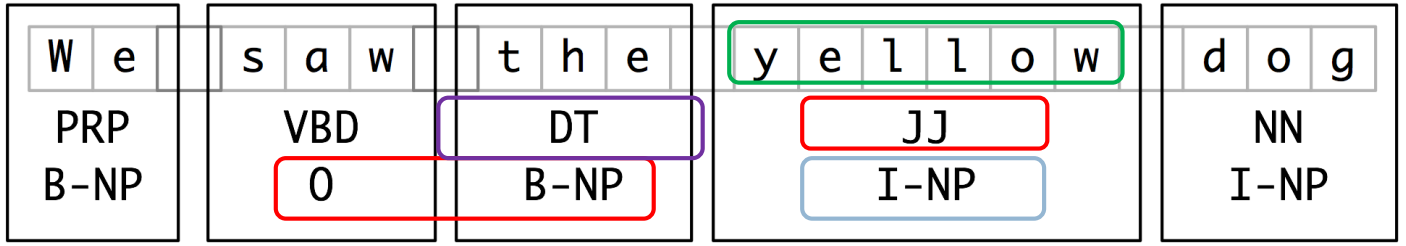
IOB-tags



- B-NP: First word in NP
- I-NP: Part of NP, not first word
- O: Not part of NP (phrase)

- Properties
  - One tag per token
  - Unambiguous
  - Does not insert anything in the text itself

Assigning IOB-tags



- The process can be considered a form for tagging
  - POS-tagging: Word to POS-tag
  - IOB-tagging: POS-tag to IOB-tag
- But one may in addition use additional features, e.g. words
- Can use various types of classifiers
  - NLTK uses a MaxEnt Classifier (=LogReg, but the implementation is slow)
  - We can modify along the lines of mandatory assignment 2, using scikit-learn

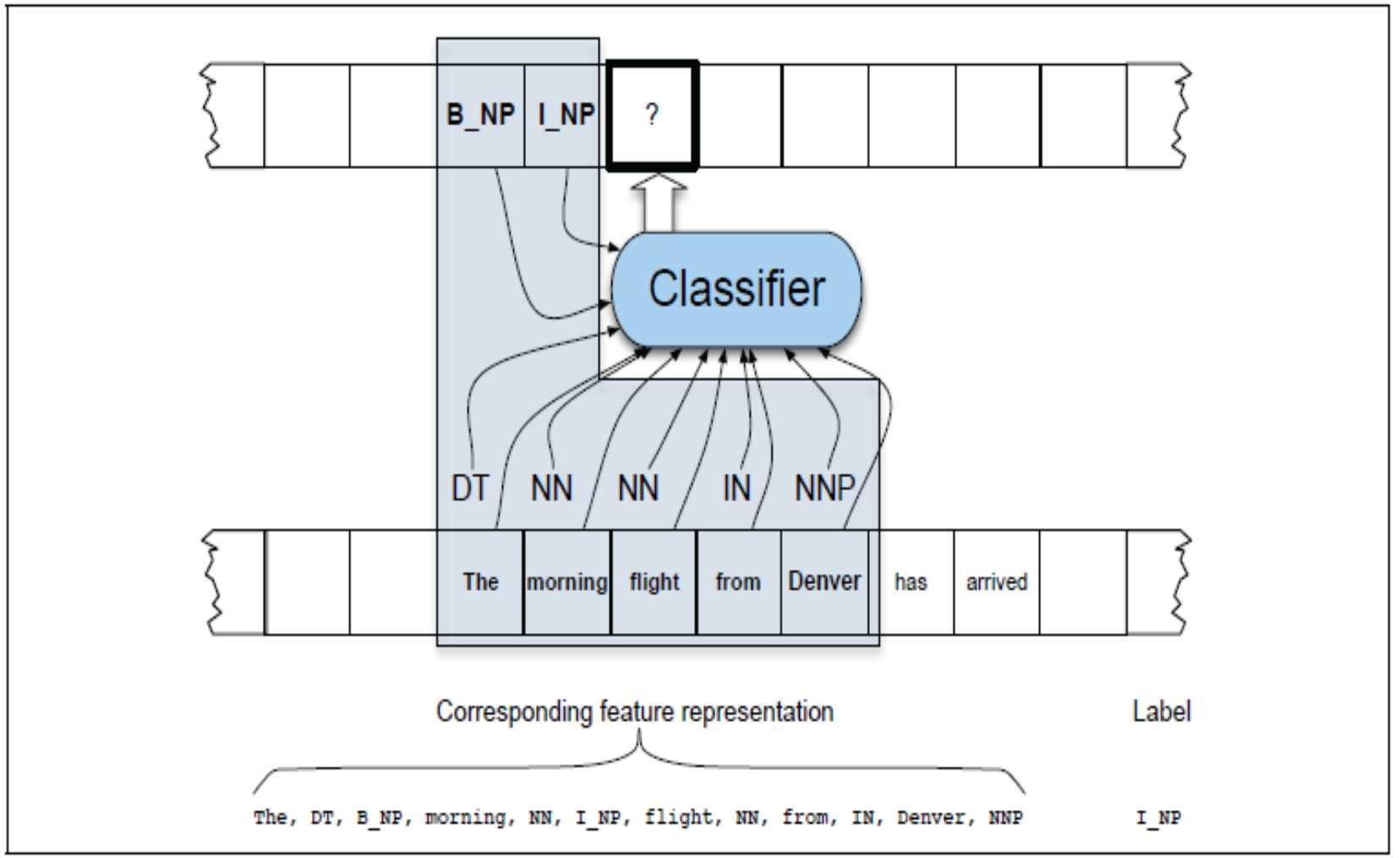


Figure 11.8 A sequence model for chunking. The chunker slides a context window over the sentence, classifying words as it proceeds. At this point, the classifier is attempting to label *flight*, using features like words, embeddings, part-of-speech tags and previously assigned chunk tags.

J&M, 3. ed.

Today

- Feedforward neural networks (partly recap)
- Recurrent networks
- Information extraction, IE
  - Chunking
- Named Entity Recognition
- Evaluation

Named entities

Citing high fuel prices, [ORG United Airlines] said [TIME Friday] it has increased fares by [MONEY $6] per round trip on flights to some cities also served by lower-cost carriers. [ORG American Airlines], a unit of [ORG AMR Corp.], immediately matched the move, spokesman [PER Tim Wagner] said. [ORG United], a unit of [ORG UAL Corp.], said the increase took effect [TIME Thursday] and applies to most routes where it competes against discount carriers, such as [LOC Chicago] to [LOC Dallas] and [LOC Denver] to [LOC San Francisco].

- Named entity:
  - Anything you can refer to by a proper name
  - i.e. not all NP (chunks):
    - *high fuel prices*
  - Maybe longer NP than just chunk:
    - *Bank of America*
- Find the phrases
- Classify them

Types of NE

Type	Tag	Sample Categories
People	PER	Individuals, fictional characters, small groups
Organization	ORG	Companies, agencies, political parties, religious groups, sports teams
Location	LOC	Physical extents, mountains, lakes, seas
Geo-Political Entity	GPE	Countries, states, provinces, counties
Facility	FAC	Bridges, buildings, airports
Vehicles	VEH	Planes, trains, and automobiles

- The set of types vary between different systems
- Which classes are useful depend on application

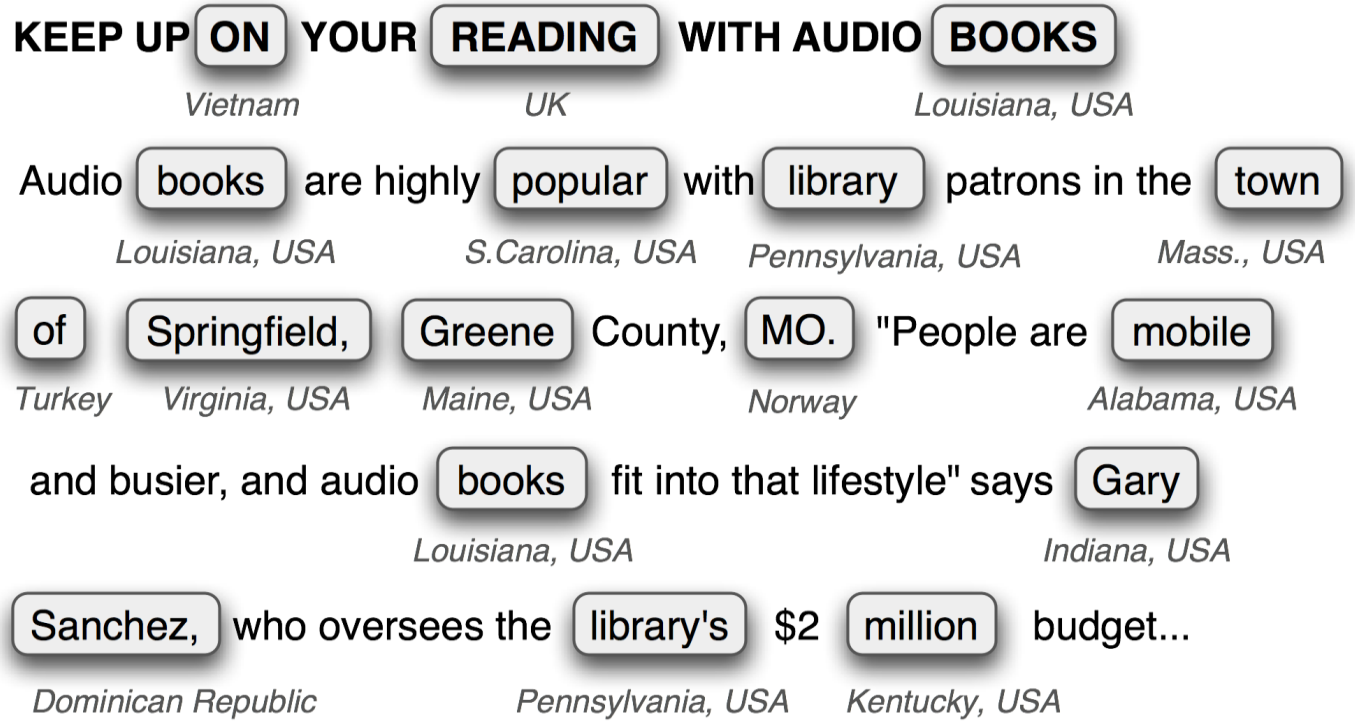
Ambiguities

Name	Possible Categories
Washington	Person, Location, Political Entity, Organization, Facility
Downing St.	Location, Organization
IRA	Person, Organization, Monetary Instrument
Louis Vuitton	Person, Organization, Commercial Product

[$_{PERS}$ Washington] was born into slavery on the farm of James Burroughs.
[$_{ORG}$ Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [$_{LOC}$ Washington] for what may well be his last state visit.
In June, [$_{GPE}$ Washington] passed a primary seatbelt law.
The [$_{FAC}$ Washington] had proved to be a leaky ship, every passage I made...

Gazetteer

- Useful: List of names, e.g.
  - Gazetteer: list of geographical names
- But does not remove all ambiguities
  - cf. example

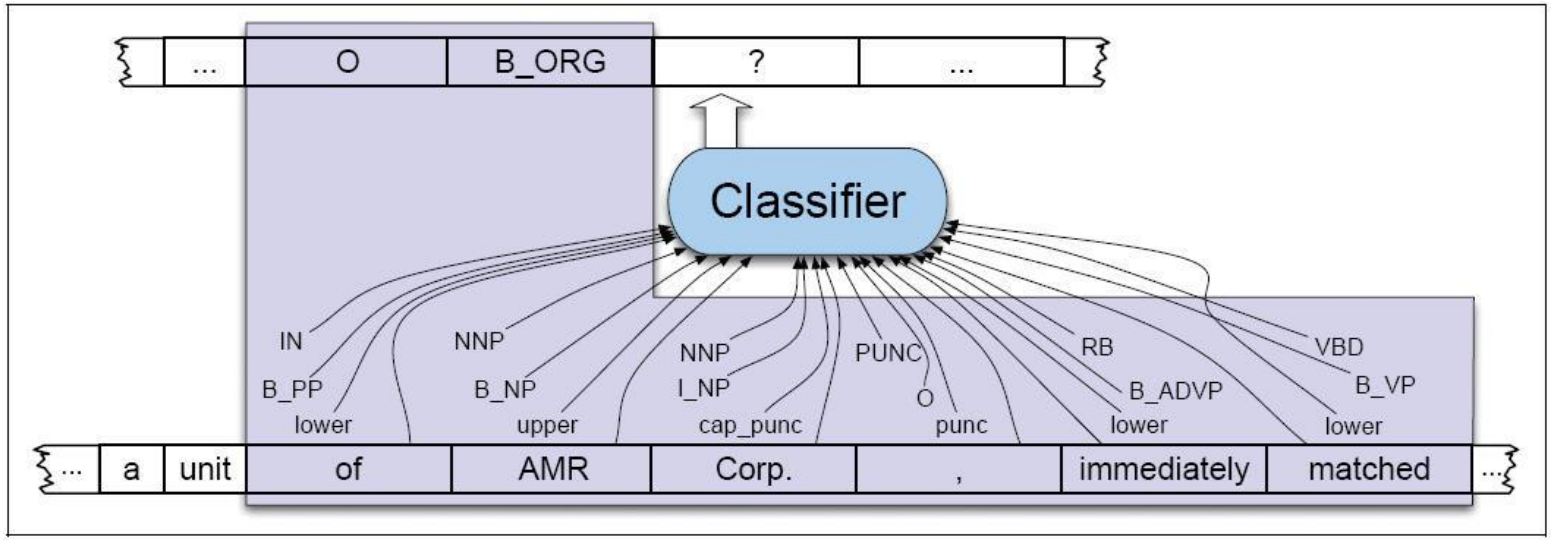


Representation (IOB)

Words	IOB Label	IO Label
American	B-ORG	I-ORG
Airlines	I-ORG	I-ORG
,	O	O
a	O	O
unit	O	O
of	O	O
AMR	B-ORG	I-ORG
Corp.	I-ORG	I-ORG
,	O	O
immediately	O	O
matched	O	O
the	O	O
move	O	O
,	O	O
spokesman	O	O
Tim	B-PER	I-PER
Wagner	I-PER	I-PER
said	O	O
.	O	O

Figure 17.4 Named entity tagging as a sequence model, showing IOB and IO encodings.

Feature-based NER



- Similar to tagging and chunking
- You will need features from several layers
- Features may include
  - Words, POS-tags, Chunk-tags, Graphical prop.
  - and more (See J&M, 3.ed)

Neural sequence labeling: NER

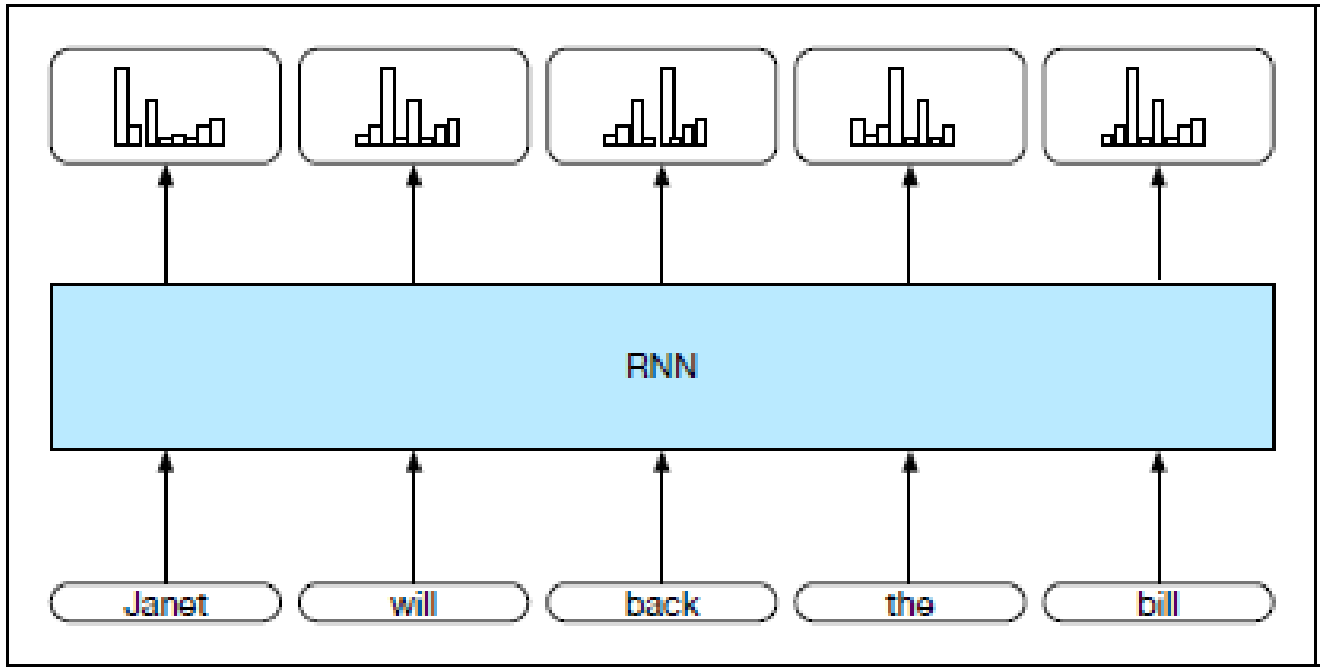


Figure 9.8 Part-of-speech tagging as sequence labeling with a simple RNN. Pre-trained word embeddings serve as inputs and a softmax layer provides a probability distribution over the part-of-speech tags as output at each time step.

From J&M, 3.ed., 2019

- We can use IOB-tags
- IOB-tagged training data
- RNN
  - Similarly to POS-tagging

A more advanced model

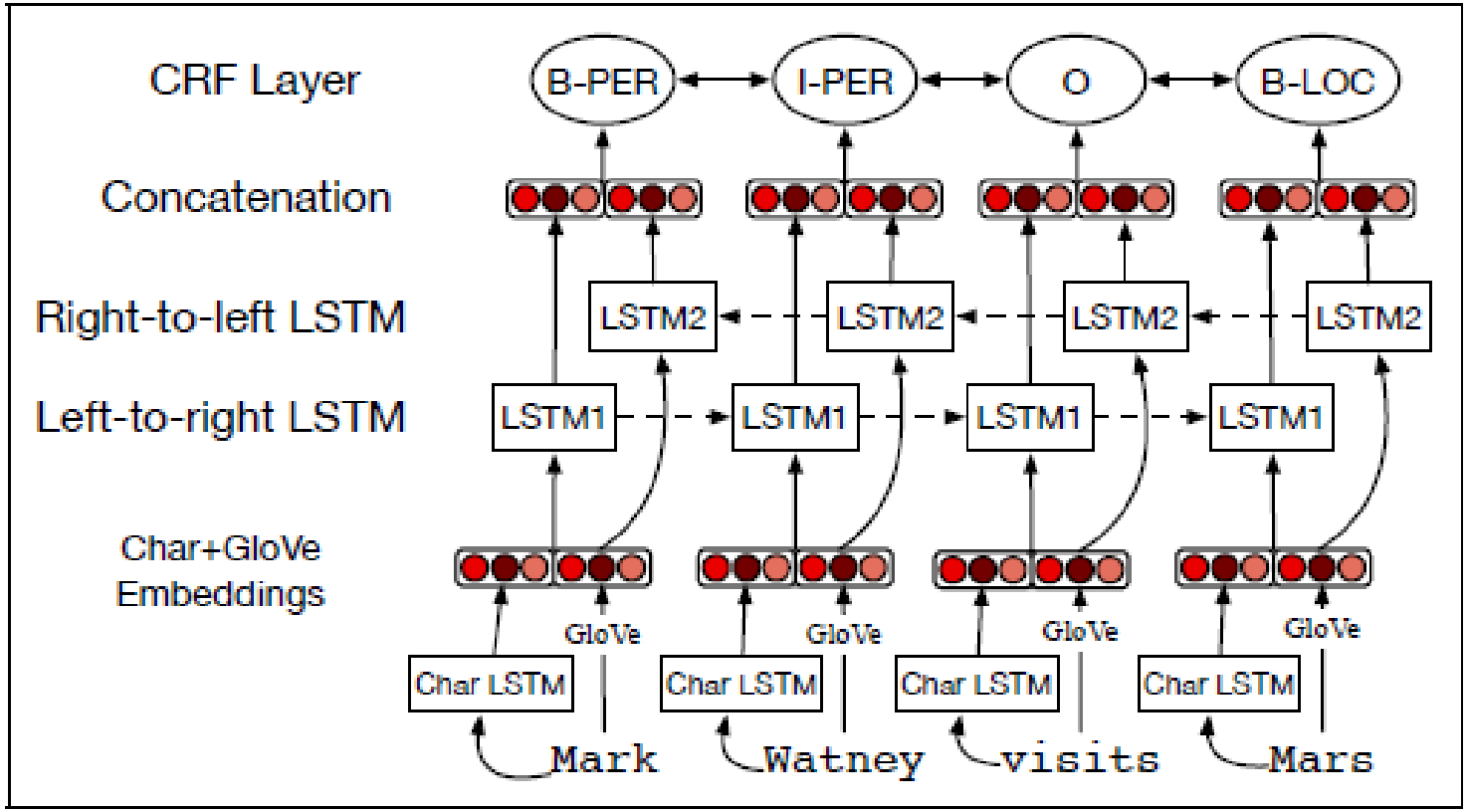


Figure 17.8 Putting it all together: character embeddings and words together a bi-LSTM sequence model. After Lample et al. (2016).

From J&M, 3.ed., 2019

- Bi-LSTM
- CRF top-layer
  - Optimize the sequence of tags
  - In contrast to optimizing individual tags (as we did it in mandatory 2)

Today

- Feedforward neural networks (partly recap)
- Recurrent networks
- Information extraction, IE
- Named Entity Recognition
- Evaluation
  - in general
  - chunkers and NER

Evaluation measure: Accuracy

- What does accuracy 0.81 tell us?
- Given a test set of 500 documents:
  - The classifier will classify 405 correctly
  - And 95 incorrectly
- A good measure given:
  - The 2 classes are equally important
  - The 2 classes are roughly equally sized
  - Example:
    - Woman/man
    - Movie reviews: pos/neg

But

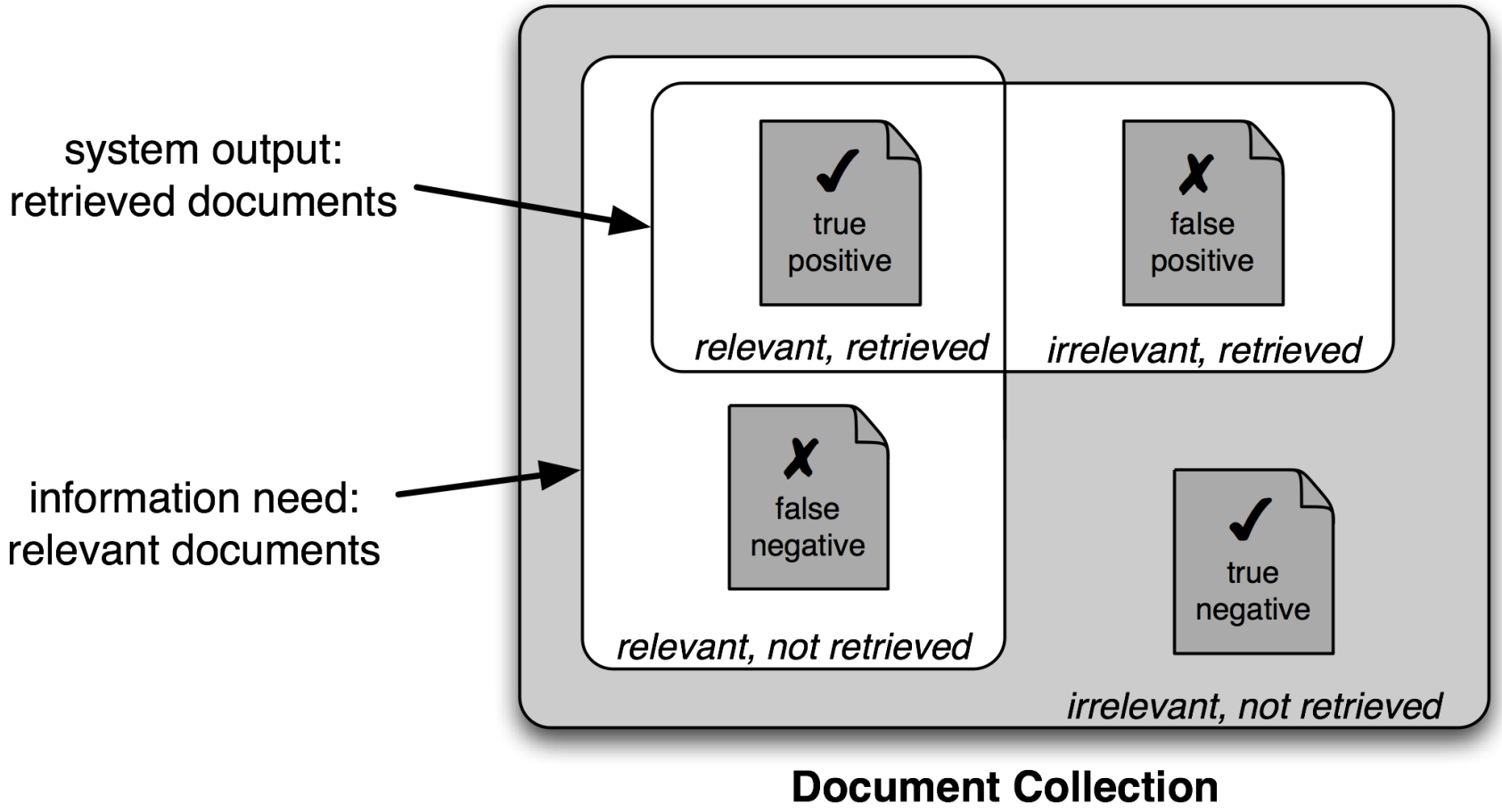
- For some tasks, the classes aren't equally important
  - Worse to loose an important mail than to receive yet another spam mail
- For some tasks the different classes have different sizes.

Information retrieval (IR)

- Traditional IR, e.g. a library
  - Goal: Find all the documents on a particular topic out of 100 000 documents,
    - Say there are 5
  - The system delivers 10 documents: all irrelevant
    - What is the accuracy?

- For these tasks, focus on
  - The relevant documents
  - The documents returned by the system
- Forget the
  - Irrelevant documents which are not returned

IR - evaluation



Confusion matrix

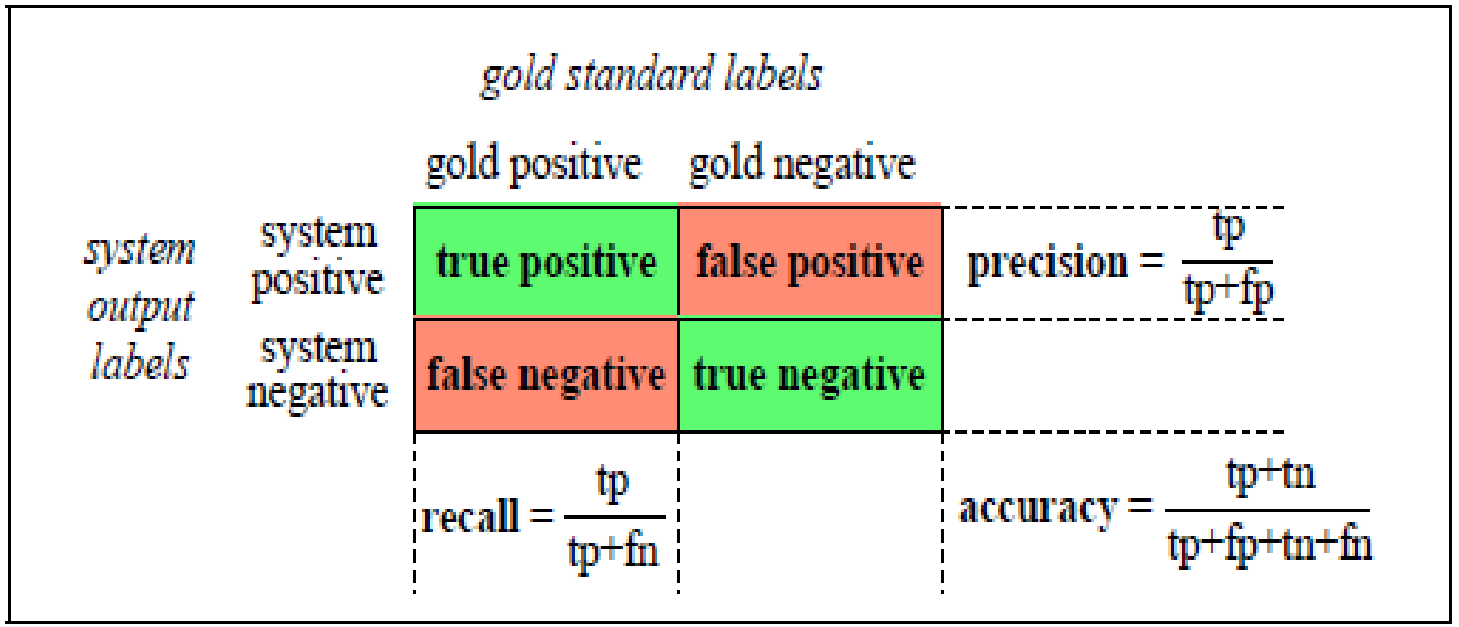


Figure 6.4 Contingency table

- Beware what the rows and columns are:
  - NLTKs ConfusionMatrix swaps them compared to this table

Evaluation measures

		Is in C	
		Yes	NO
Classifier	Yes	tp	fp
	No	fn	tn

- Accuracy: (tp+tn)/N
- Precision:tp/(tp+fp)
- Recall: tp/(tp+fn)

- F-score combines P and R
- $F_1 = \frac{2PR}{P+R} \left(= \frac{1}{\frac{\frac{1}{R}+\frac{1}{P}}{2}} \right)$
- F_1 called "harmonic mean"
- General form
  - $F = \frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}}$
  - for some $0 < \alpha < 1$

Confusion matrix

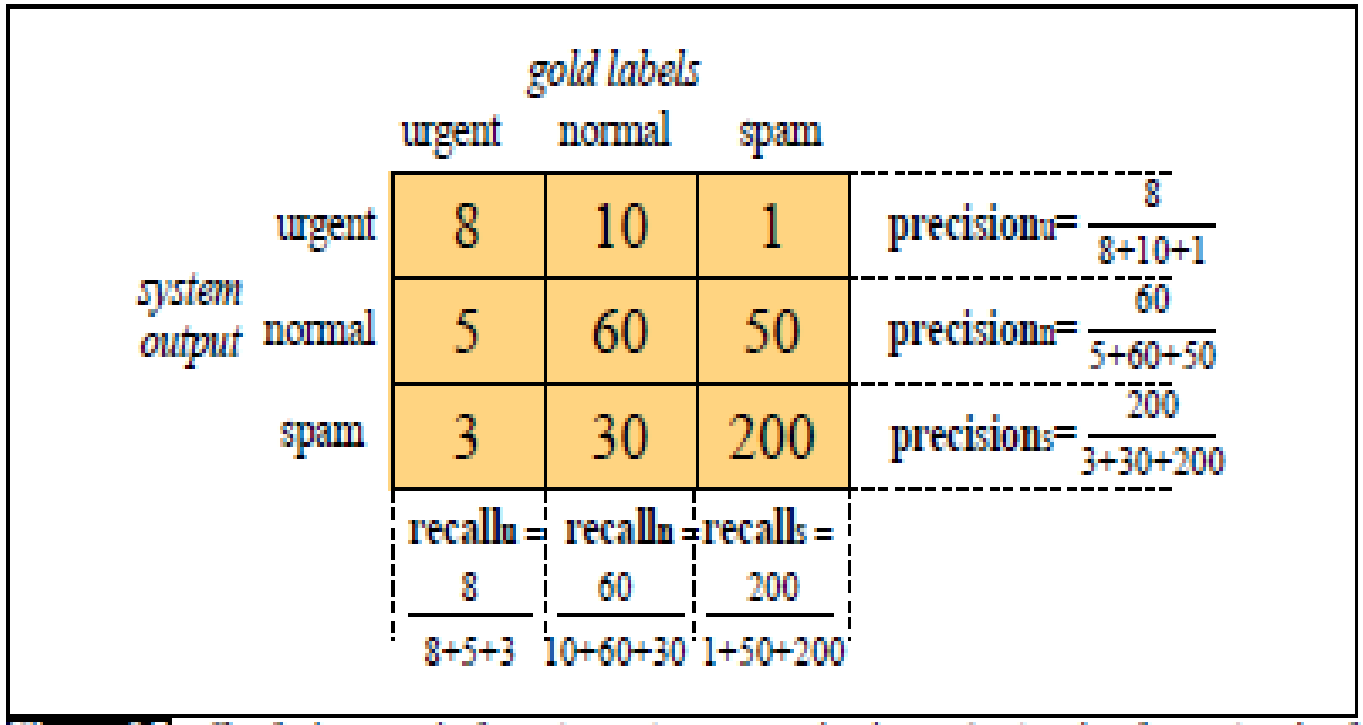


Figure 6.5 Confusion matrix for a three-class categorization task, showing for each pair of classes (c_1, c_2), how many documents from c_1 were (in)correctly assigned to c_2

- Precision, recall and f-score can be calculated for each class against the rest

Today

- Feedforward neural networks (partly recap)
- Recurrent networks
- Information extraction, IE
- Named Entity Recognition
- Evaluation
  - in general
  - chunkers and NER

Evaluation

- Have we found the correct NERs?
  - Evaluate precision and recall as for chunking
- For the correctly identified NERs, have we labelled them correctly?

Evaluating (IOB-)chunkers

- cp = nltk.RegexpParser("")
- test_sents = conll ('test', chunks=['NP'])
- IOB Accuracy: 43.4%
- Precision: 0.0%
- Recall: 0.0%
- F-Measure: 0.0%

- What do we evaluate?
  - IOB-tags? or
  - Whole chunks?
  - Yields different results
- For IOB-tags:
  - Baseline:
    - majority class O,
    - yields > 33%
- Whole chunks:
  - Which chunks did we find?
  - Harder
  - Lower numbers

Evaluating (IOB-)chunkers

- cp = nltk.RegexpParser("")
- test_sents = conll ('test', chunks=['NP'])
- IOB Accuracy: 43.4%
- Precision: 0.0%
- Recall: 0.0%
- F-Measure: 0.0%

>> cp = nltk.RegexpParser(r"NP: {<[CDJNP].*>+}")

- IOB Accuracy: 87.7%
- Precision: 70.6%
- Recall: 67.8%
- F-Measure: 69.2%

In	IN	O	O
addition	NN	B-NP	B-NP
to	TO	O	O
his	PRP$	B-NP	B-NP
previous	JJ	I-NP	I-NP
real-estate	NN	I-NP	I-NP
investment	NN	I-NP	I-NP
and	CC	I-NP	I-NP
asset-management	NN	I-NP	I-NP
duties	NNS	I-NP	I-NP
,	,	O	O
Mr.	NNP	B-NP	B-NP
Meador	NNP	I-NP	I-NP
takes	VBZ	O	O
responsibility	NN	B-NP	B-NP
for	IN	O	O
development	NN	B-NP	B-NP
and	CC	O	I-NP
property	NN	B-NP	I-NP
management	NN	I-NP	I-NP
.	.	O	O

Next week

- Relation extraction (sec. 17.2)
- Encoder-Decoder Models (sec. 10.1-10.2)